What is claimed is:

1. A method of congestion control in transmission of data in packets over a network link using a transport layer protocol, wherein:

   a) the number of unacknowledged packets in transit in the link is less than or equal to a congestion window value $cwnd_i$;

   b) the value of $cwnd_i$ is varied according to an additive-increase multiplicative-decrease (AIMD) law having an increase parameter α_i, and

   c) the value of α_i is increased during each congestion epoch.

2. A method of congestion control according to claim 1 in which the value of α_i increases at a fixed time after the start of each congestion epoch.

3. A method of congestion control according to claim 2 in which the fixed time is calculated as a fixed multiple of the round-trip time for a data packet to travel over the network link.

4. A method of congestion control according to claim 1 in which the value of α_i increases at a plurality of fixed times after the start of each congestion epoch.

5. A method of congestion control according to claim 4 in which each fixed time is calculated as a respective fixed multiple of the round-trip time for a data packet to travel over the network link.

6. A method of congestion control according to claim 1 in which the value of α_i is unity at the start of each congestion epoch.

7. A method of congestion control according to claim 1 in which the value of α_i increases as a function of time from the start of a congestion epoch.

8. A method of congestion control according to claim 7 in which α_i increases as a polynomial function of time from the start of a congestion epoch.

9. A method of congestion control according to claim 1 in which, upon detection of network congestion during a kth congestion epoch at a time when the value of $cwnd_i$ is $w_i(k)$, the value of $cwnd_i$ becomes $\beta_i w_i(k)-\delta$ where $\delta = 0$ initially and $\delta_i = \beta_i(\alpha_i^H - \alpha_i^L)$ after an increase in the value of α_i.

10. A method of transmitting data in packets over a network link in which network congestion is controlled by a method according to claim 1.

11. A method according to claim 10 in which during each congestion epoch, at a time prior to increase in the value of α_i, the method implements the transport control protocol (TCP) having standard congestion control.

12. A networking component for transmission of data in packets over a network link using a transport layer protocol, the networking component being operative to implement congestion control, wherein:

a) the number of unacknowledged packets placed by the networking component in transit on the link is less than or equal to a congestion window value $cwnd_i$;

b) the value of $cwnd_i$ is varied according to an additive-increase multiplicative-decrease (AIMD) law having an increase parameter α_i; and

c) the value of α_i is increased during each congestion epoch.

13. A networking component according to claim 12 in which the value of α_i is increased at a fixed time after the start of each congestion epoch.

14. A networking component according to claim 13 in which the fixed time is calculated as a fixed multiple of the round-trip time, being the interval between the networking component placing the packet on the network link and its receiving an acknowledgement of receipt of the packet.

15. A networking component according to claim 12 in which the value of α_i is increased at a plurality of fixed times after the start of each congestion epoch.

16. A networking component according to claim 15 in which each fixed time is calculated as a respective fixed multiple of the round-trip being the interval between the networking component placing the packet on the network link and its receiving an acknowledgement of receipt of the packet.

17. A networking component according to claim 12 in which the value of α_i is unity at the start of each congestion epoch.

18. A networking component according to claim 12 in which the value of α_i is increased as a function of time from the start of a congestion epoch.

19. A networking component according to claim 18 in which α_i is increased as a polynomial function of time from the start of a congestion epoch.

20. A networking component according to claim 12 which operates, upon detection of network congestion during a *k*th congestion epoch and at a time when the value of $cwnd_i$ is $w_i(k)$, to modify the value of $cwnd_i$ to $\beta_i w_i(k)-\delta$ where $\delta = 0$ initially and $\delta_i = \beta_i(\alpha_i^H - \alpha_i^L)$ after an increase in the value of α_i, β_i being a decrease parameter.

21. A networking component according to claim 12 implemented in executable computer code.

22. A method of congestion control in transmission of data in packets over a network link using a transport layer protocol, wherein:

    a) the number of unacknowledged packets in transit in the link is less than or equal to a congestion window value $cwnd_i$;

    b) the value of cwnd is varied according to an additive-increase multiplicative-decrease (AIMD) law having a multiplicative decrease parameter β_i, and

    c) the value of β_i is set as a function of one or more characteristics of one or more data flows carried over the network link.

23. A method of congestion control according to claim 22 in which the value of β_i is set as a function of the round-trip time of data traversing the link.

24. A method of congestion control according to claim 23 in which the link carries a plurality of data flows, there being a round-trip time RTT_i associated with the ith data flow sharing the link, the shortest round-trip time being designated $RTT_{min,i}$ and the greatest round-trip time being designated $RTT_{max,i}$, wherein the value of β_i is set as $\beta_i = \frac{RTT_{\min,i}}{RTT_{\max,i}}$.

25. A method of congestion control according to claim 24 in which the values of $RTT_{min,i}$ and $RTT_{max,i}$ are monitored and the value of $\beta_i = \frac{RTT_{\min,i}}{RTT_{\max,i}}$ is re-evaluated periodically.

26. A method of congestion control according to claim 22 in which the additive-increase multiplicative-decrease law has an increase parameter α_i, and α_i is varied as a function of β_i.

27. A method of congestion control according to claim 26 in which and α_i is varied as $\alpha_i = 2(1 - \beta_i)$.

28. A method of congestion control according to claim 22 in which the value of round-trip times of one or more data flows carried over the network link are monitored periodically during transmission of data and the value of β_i is adjusted in accordance with updated round-trip values thereby determined.

29. A method of congestion control according to claim 22 in which the value of β_i is set as a function of the mean inter-packet time of data flowing in the link or of the mean throughput.

30. A method of congestion control according to claim 22 in which the value of β_i is set by:

a) during data transmission, periodically monitoring the value of the mean inter-packet time $IPT_{min,i}$ or throughput of the i'th flow;

b) upon the measured value of $IPTmin$ moving outside of a threshold band, resetting the value of β_i to $\beta_{reset,i}$; and

c) upon $IPTmin,i$ or throughput returning within the threshold band, setting $\beta_i = \frac{RTT_{\min,i}}{RTT_{\max,i}}$ and periodically resetting β_i in response to changes in the value of $RTTmin,i$ or $RTTmax,i$.

31. A method of transmitting data in packets over a network link in which network congestion is controlled by a method according to claim 30.

32. A networking component for transmission of data in packets over a network link using a transport layer protocol, the networking component being operative to implement congestion control, wherein:

a) the number of unacknowledged packets placed by the networking component in transit on the link is less than or equal to a congestion window value $cwnd_i$;

b) the value of $cwnd_i$ is varied according to an additive-increase multiplicative-decrease (AIMD) law having a multiplicative decrease parameter β_i; and

c) the value of β_i is set as a function of one or more characteristics of one or more data flows carried over the network link.

33. A networking component according to claim 32 in which the value of β_i is set as a function of the round-trip time, being the interval between the networking component placing a packet on the network link and its receiving an acknowledgement of receipt of the packet.

34. A networking component according to claim 33 operative to transmit a plurality of data flows on the link, there being a respective round-trip time RTT_i associated with the ith data flow sharing the link, the shortest round-trip time being designated $RTT_{min,i}$ and the greatest round-trip time being designated $RTT_{max,i}$, wherein component sets the value of β_i as $\beta_i = \frac{RTT_{\min,i}}{RTT_{\max,i}}$ ..

35. A networking component according to claim 34 operative to determine the values of $RTT_{min,i}$ and $RTT_{max,i}$ and re-evaluate the value of β_i periodically.

36. A networking component according to claim 35 that calculates the value of $RTT_{max,i}$ from the value of β_i during previous congestion epochs.

37. A networking component according to claim 33 in which the additive-increase multiplicative-decrease law has an increase parameter α_i, and which is operative to vary α_i as a function of β_i.

38. A networking component according to claim 37 that varies α_i as $\alpha_i = 2(1 - \beta_i)$.

39. A networking component according to claim 33 that monitors periodically during transmission of data the value of round-trip times of one or more data flows that it implements on a network link and adjusts the value of β_i in accordance with updated round-trip values thereby determined.

40. A networking component according to claim 39 that sets the value of β_i as a function of the mean inter-packet time of data flowing in the link.

41. A networking component according to claim 40 that sets the value of β_i is set as a function of the minimum of the mean inter-packet time ($IPT_{min,i}$), where the mean is taken over a round-trip time period, being the interval between the networking component placing a packet on the network link and its receiving an acknowledgement of receipt of the packet.

42. A networking component according to claim 41 that sets the value of β_i by:

a) during data transmission, periodically monitoring the value of the mean inter-packet time IPT_{min} or the mean throughput;

b) upon the measured value of IPTmin or the mean throughput moving outside of a threshold band, resetting the value of β_i to β_{reset}; and

c) upon IPT_{min} or the mean throughput returning within the threshold band, setting $\beta_i = \frac{RTT_{min,i}}{RTT_{max,i}}$ and periodically resetting β_i in response to changes in the value of $RTTmin,i$ or $RTTmax,i$ or the mean throughput.

43. A networking component according to claim 33 implemented in executable computer code.